(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 491 348 A2

(12) # EUROPEAN PATENT APPLICATION

(21) Application number: 91121582.0

(22) Date of filing: 17.12.91

(51) Int. Cl.5: G07G 1/00, A47F 9/04, G07F 7/00

(30) Priority: 17.12.90 FI 906202

(43) Date of publication of application: 24.06.92 Bulletin 92/26

(84) Designated Contracting States: BE DE DK FR GB NL SE

(71) Applicant: Mannerjoki, Mika
Metsälammintie 3 A
SF-05460 Hyvinkää(FI)
Applicant: Purosto, Tero
Välenojankatu 19
SF-05880 Hyvinkää(FI)

(72) Inventor: Mannerjoki, Mika
Metsälammintie 3 A
SF-05460 Hyvinkää(FI)
Inventor: Purosto, Tero
Välenojankatu 19
SF-05880 Hyvinkää(FI)

(74) Representative: Zipse + Habersack
Kemnatenstrasse 49
W-8000 München 19(DE)

(54) **Unstaffed checkout system.**

(57) The invention relates to an unstaffed checkout system comprising a checkout counter (1) provided with appropriate actuators, a background system (17) containing at least the product data, and a cash terminal program (15) which transmits information from the checkout counter actuators to the background system. The system comprises a functional block (13) with an interface (14) compatible with the data transmitted by the cash terminal program (15). Product data relating to product identification are collected and new data are automatically added to the data for each product until the system has learned the key characteristics of each product.

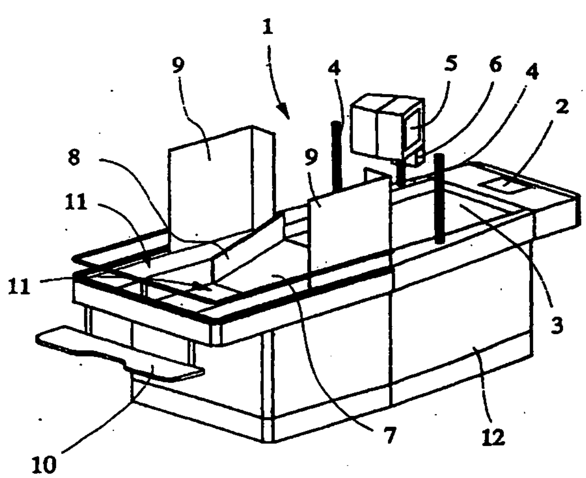

Fig. 1

EP 0 491 348 A2

The present invention relates to an unstaffed checkout system comprising a checkout counter provided with appropriate actuators, a background system containing at least the product data, and a cash terminal program which transmits information from the checkout counter actuators to the background system.

One of the previously known checkout systems is presented in patent application no. GB 2161631. In this solution, one salesperson takes care of collecting the money from the customers, who, using the checkout counter equipment, feed the id ntification data for the products they have bought into the cash system of the shop. After all the products bought have been presented to the checkout counter in an acceptable manner, the customer may proceed further and receives a receipt on the basis of which he/she is then charged by the salesperson. A problem in this prior-art solution is that it still requires a salesperson to collect the money and a space for him/her to work in. This space naturally reduces the effective area of th shop. Moreover, the product identification data must be updated separately every time new products are brought for sale or the data for old products are changed. The updating process involves a risk of error because it may be neglected and because the data may be incorrectly put into the system. A further drawback is that this system cannot be flexibly added to existing checkout systems but replaces the whole existing system. Consequently, this system is expensive and inflexible. Y t another serious drawback is that file updates are not effected via a background system, so each ch ckout counter must contain its own product data and product identification data until the data are transferred manually or otherwise separately into the other machines. As a result, the machines practically always have slightly different data and th system cannot be self-learning.

The object of the present invention is to eliminate the drawbacks referred to above and to achieve a reliable, cheap and self-updating unstaffed checkout system. The invention is characterized by what is presented later on in the claims. The invention has the advantage of reducing the need for sales personnel and providing easy connection to existing checkout systems. Via an interface, a product identification program constituting a separate functional unit is linked by software with an existing cash terminal program. The interface can be so standardiz d that the product identification program can be flexibly linked with different cash terminal programs as the interface data ar always compatible and the cash terminal program knows what data is transmitted by the product identification program and vice versa. When the interface is a variable, variable address or memory location mutually agreed on by the program suppliers, the result is an extremely cheap and usable interface. A further advantage is the self-learning capability of the system, which means that n w product identification data or changes in the old data need not necessarily be updated for the machine but the system learns the new data itself.

In the following, the invention is described by the aid of an example by referring to the attached drawings, in which

Fig. 1 presents an overall view of the checkout counter of the invention in a perspective drawing,

Fig. 2 presents a block diagram of an embodiment of the system of the invention, showing how the actuators are connected to the system,

Fig. 3 presents a functional block diagram of representing the recognition of a bank or other card and the input of the starting data,

Fig. 4 presents a functional block diagram of the functions of the scanning block of the checkout counter,

Fig. 5 presents a functional block diagram of the functions of the learning block of the checkout counter,

Fig. 6 presents a functional block diagram of the functions of the EAN-code block of the checkout counter,

Fig. 7 presents a functional block diagram representing the handling of errors.

Fig. 1 illustrates the checkout counter 1 of the invention. Depending on the need, the shop is provided with one or more checkout counters, which are placed e.g. side by side with a passage between them leading out of the shop space. The passage is blocked by a gate, which is opened after an accepted transaction. Each checkout counter is connected to the electric network and, by means of a connecting cable, to the shop's central computer, which runs the background system. The connections to the electric network and the central computer are implemented using conventional techniques, so they are not presented in the drawings. At the entry end of the checkout counter is a scanner 2, whose function is to recognize all products marked with a bar code. The customer him/herself passes the products he/she has bought over the scanner. After the scanning, the customer places the product on a belt conveyor, which is provided with a weighing device for determining the weight of the products. The weigher is placed under the belt conveyor and is not visible in the drawings. Similarly, the belt conveyor actuator is not visible in the drawings because it is placed inside the checkout counter. In the direction of motion of the belt conveyor, at the middle part of it,

there is a light screen 4 for the recognition of the shape of the products. The light screen extends from the level of the conveyor belt to a height sufficient to allow the recognition of all everyday products regardless of their height. At the entry end of the checkout counter there is also a monitor 5 through which the customer is given the necessary instructions, and also a bank card reader and a PIN keypad 6, by means of which the customer can pay for the goods purchased. The belt conveyor 3 delivers the products onto a trough belt conveyor 7 placed directly after it on the checkout counter. The trough belt conveyor moves the accepted products into one of the troughs 11 provided at the exit end of the checkout counter as guided by the distributor 8. The actuator of the trough belt conveyor 7 and the turning mechanism of the distributor 8 are placed inside the checkout counter and are not visible in the drawings. Both side edges of the checkout counter are provided with protective walls 9 placed alongside the trough belt conveyor 7 to prevent the products from falling off the counter. The protective walls 9 also prevent the customer from throwing any products past the light screen into the troughs 11. At the extremity of the exit end of the checkout counter there is a packing board 10 for ease of packing the products into a shopping bag. Moreover, inside the checkout counter there is a computer which is provided with all the necessary interface cards and which is used to run the product identification program 13, allowing unstaffed operation of the checkout counter. The actuators and the computer 16 inside the checkout counter are protected by the counter body, which is covered with side walls 12.

Fig. 2 is a simplified presentation of an embodiment of the system of the invention, showing the main blocks of the product identification program 13 as well as the connections between the various actuators used in the checkout system on the one hand and the product identification program and the computer on the other hand. The broken line marked with reference number 1 in Fig. 2 represents the checkout counter and the broken line marked with reference number 16 represents the microcomputer physically present in the checkout counter and used to run a program consisting of the product identification program 13 and the cash terminal program 15. The product identification program 13 and the cash terminal program 15 are linked together via an interface 14 so as to allow interactive operation. The interface 14 is like a mailbox in which eith r program block may store data to be read by the other block. For each data type there is a separate interface agreed on by the program suppliers, so the program which writes or reads data in an interface knows automatically the type of data to be written or read in each interface.

The data types referred to include scanning start permission, credit card code inquiry, EAN data, acceptance data, keyboard input data and file updates. In practice, the interface 14 is a series of variables, variable addresses or memory locations in the computer memory.

The card reader and PIN keypad 6 is plugged into the computer's keyboard connector. This actuator is only used to supply input data to th cash terminal program, this being indicated in Fig. 2 by an arrow pointing from the actuator block towards the cash terminal program block 15. The bar code reader or scanner 2 is connected to the COM2 port of the computer, the weigher to the COM3 port and the background system 17 run in the central computer of the shop is connected to the COM1 port of the checkout counter computer. The modem, which is used for the transmission of advertisements and maintenance data, is connected to the COM4 port of the checkout counter computer. The bar code reader and the weigher supply only input data to the product identification program 13, whereas the background system and the cash terminal program 15 may communicate in both directions. Similarly, a two-way data transmission link is provided between the modem and the product identification program. All the above-mentioned COM ports are standardized serial ports connected to the serial communication bus of the computer.

The receipt printer is connected to a parallel port, e.g. a Centronics interface, and the monitor 5 is connected to a VGA port. The rest of th actuators are connected to the checkout counter computer via a separate I/O card provided with input pins for the data supplied by the light scre n 4 for shape recognition, by the PIN-keypad and for various sensor data. These sensor data include trough-empty data for the right-hand and left-hand troughs, data indicating the final limit of the belt conveyor 3 and data indicating the right-hand and left-hand limits of the distributor 8. Moreover, th same I/O card is provided with pins for the output data needed by the control electronics for the control of the belt conveyor 3, the trough belt conveyor 7, the turning device of the distributor 8 and the lock of the customer gate. The customer gate, the function of which is to open the passag for the customer to exit from behind the checkout counter after the products have been accepted and paid for, is not shown in the drawings. The communication 18 between different checkout counters, mainly consisting of the transmission of updated product and product identification data, takes place via the background system 17. In the mass storage of the background system there is a file containing the product data, of which certain items, such as the EAN code, the price and the product name, are transmitted via the cash t rminal program 15 into a

product file in the mass storage of the product identification program 13. In this same file, the product identification program collects product identification data, such as product weight, height etc., to b used in the learning process and transmitted via the background system to the other checkout counters. This provides the advantage that the learning routine need not be repeated separately in each checkout counter.

Fig. 3 shows a functional block diagram representing the reading of the bank card. For the sake of clarity, the text appearing on the monitor screen is presented in separate blocks and enclosed in quotes. When a customer comes to the checkout counter, he/she starts the product identification and payment procedure by first communicating with the customer terminal. First, the customer terminal wants to read the bank card or equivalent and asks the customer to insert his/her card into the card reader. Depending on the actual equipment used, the card is either inserted into the machine or just slid past its read head. The monitor displays the appropriate text in each case. Next, the program prompts the customer for a personal code number and performs a comparison to check whether the number given is correct. If the number was not right, the program outputs a corresponding message via the monitor and asks the customer to input his/her personal code number again. After the correct number has been input, the program can proceed further and asks the customer to select the manner of payment. In practice, there are three alternative ways: bank card, credit card or a combination of these. After the customer has typed in his selection for the manner of payment, the program contacts the background system and compares the customer data against a checklist in the background system to see if the customer's card and payment mode selection are acceptable. If the card or the payment mode was uncacceptable, a corresponding message is output via the monitor and the customer is prompted to remove the card if it is still in the reader. At the same time, the program flow returns to the beginning and the customer is asked to insert another card. If the payment mode selected was acceptable, the monitor displays a message asking the customer to r move the card if it is still in the reader and the program gives a permission to start the transaction. Next, the program checks whether one of the troughs 11 at the exit end of the checkout counter is empty. When either one of the troughs is empty, the checkout counter can be started, the distributor 8 is turned to the appropriate position and a suitable instruction, .g. "START SCANNING", is displayed by the monitor. After this, program control is handed over to the functional block shown in Fig. 4.

Fig. 4 presents the scanning block, i.e. the bar code reading block. The basic function in this block consists in the program waiting until scanning has taken place. At this stage, it is important for the system to be able to ensure that no products are moved directly past the scanner, the weigher or the light screen. Also, the program checks that there are not too many products at the same time on the belt conveyor and that a new product can only be scanned after the previous one has been weighed. First, the program checks whether a product has been scanned. If no scanning has taken place, the program reads the weight from the weigher and then checks whether a change has occurred in the light screen. If no change has taken place, the program checks whether the weight indicated by the weigher has changed. If there is no change in the weight, either, the program checks whether a wait flag has been set. A wait flag must be set for the shape recognition at the light screen because the scanning and weighing are very fast operations, whereas moving the product through the light screen takes a much longer time. Setting the wait flag enables the system to know that measurement is still going on at the light screen, so the system will avoid producing incorrect data or error states. While measurement is going on at the light screen, one of the photocell connections is still blocked. If the wait flag was not set, the action returns to the beginning, i.e. the program continues waiting for scanning.

On the other hand, if the customer has scanned a product, a corresponding scanning message is transmitted via a serial port to the product identification program. The EAN code of the product is now read into register E, which contains the running number of the product scanned. Moreover, the value of register E is increased by one to enable the system to know how many products have been scanned and how many products are to be expected to arrive to the weigher. The first product that was brought onto the weigher must also be the first to be removed from it, otherwise an error has occurred. Next, the program checks the product file to see if it contains the EAN code of the product. This function corresponds to the code inquiry function shown in Fig. 2. If the EAN code is not found in the background system of the shop, the cash terminal program in the checkout counter cannot use the EAN code in question, and in this case the checking operation results in an ERROR function, which is presented as a block diagram in Fig. 7. When the ERROR function is activated, a corresponding error message is output via the monitor 5 and the belt conv yor 3 is returned to the starting position. Next, a message or alarm signal is sent to the background room of the shop to allow the error to be acknowledged, i.e. the

missing EAN code to be added to the background system. For each product, the following data items are written into the product file: EAN code, product name and price. Moreover, via the learning file, the system adds automatically certain product identification data, such as weight, height, etc., to the product file. The computer in each checkout counter has a product file in its mass storage, i.e. on its hard disk. Upon an alarm, the new product may be added via a touch-screen e.g. by the shop supervisor. After the addition, the program returns back to the scanning block, to the function from which the ERROR function was started. If and when the EAN code is found in the product file, the program again continues monitoring the weigher and the light screen. For example, the customer may attempt to throw the product through the light screen so that it will not land on the weigher. Suppose the product has been placed in the normal manner on the belt conveyor 3, where it is weighed immediately. As yet, no change has taken place in the light screen, so the next check is whether a change has occurred in the weight indicated by the weigher. Since the product is on the belt conveyor 3, a change in the weight has occurred, and in this case a new check is performed to ascertain whether the product has been removed from the weigher. If the equation weight(meas) = weight(prev) - weight(1) becomes true, this means that product(1) has been removed from the weigher. In this equation, weight(meas) is the value indicated by the weigher at the time of measurement, weight(prev) is the previous value measured by the weigher and weight(1) is the weight of the first product in the queue, i.e. of the product which is to he the first to be removed from the weigher. After this, the weigher register is decreased by one, weight(prev) is set to the value of weight(meas), i.e. the weigher is set for the product in question. Next, a check for the light screen is performed to establish whether the wait flag has been set, i.e. whether a photocell has been lit. In a normal situation, product(1) has not yet passed through the light screen at this stage, so the correct result of the checking operation must be "no", otherwise an error has occurred. If the equation weight(meas) = weight(prev) - weight(1) does not become true during the above-mentioned check, the program must check whether the EAN code has been read into register E. If in this situation the EAN code has not been read into register E, this means that no scanning has yet been performed but a change has nevertheless occurred in the weight indication. This produc s an error situation and activates the ERROR fucntion. Similarly, if the equation weight(meas) = weight(prev) - weight(1) does not become true in the above-mentioned checking situation but the latter check indicates that the EAN code has been r ad into register E, the program proceeds to block A in the teaching procedur presented in Fig. 5. In this procedure, error recognition takes place, which is required to ensure that the data learned are correct.

First, the system checks whether the weight of the product scanned is correct. If it is, program execution continues from the light screen block (LSBLK), which is presented in Fig. 4. In this block, the system first sets wieght(n) = weight(meas)-weight(prev), where n is the running number of the product being weighed. By means of this number it is possible during continuous weighing to d tect the weight of each separate product brought onto the weigher even though there is a continuous flow of products arriving to, staying on and being r - moved from the weigher. Next, the value of the weigher register is increased by one and the weigher is reset, weight(prev) is set to the value of weight(meas), whereupon the program returns to the scanning block to check whether a change has taken place in the light screen.

If in the learning block presented in Fig. 5 the weight of the product scanned was not correct, then it is necessary to check whether the numb r of samples is full. The number of samples is a certain preselected number which ensures a good learning result with the required probability. The number of samples may have values e.g. between 0-9. If the number of samples was not full, the weight is accepted, the product identification data is stored in the learning data file and program execution continues again from the light screen block (LSBLK), whereafter execution is back in the normal scanning block. Similarly, if the same kind of product has been passed through the weigher so many times that the number of samples becom s full at this stage of the learning block, the result is an error situation because there is now something wrong with the weight. In a normal situation this learning phase would not have been needed. A typical fault leading to this situation could be e.g. that a one-litre milk carton has lost so much milk through leakage that its weight is no longer within the tolerance range. In this situation the system helps the customer get a good product of the proper weight.

Now, let us consider the learning block and the passage of the product through the light screen. In Fig. 5, the relevant block is SCBLK-B, to which the program jumps from the scanning block when a change has occurred in the light scr n and th lowest light cell is off. In this case, the program first resets the wait flag, then reads the light screen into register m, where m is the running number of the product under measurement, and increases the height register value by one. Next, it ch cks whether height x, y or z is equal to the height of product-

(1). Product(1) again stands for the first product in the queue. If this check yields a positive result, the program then checks whether the number of samples is full. If it is, the height register value is decreased by one and execution is continued from the EAN block, which is presented in Fig. 6. If the height check gives a negative result, the program again checks whether the number of samples is full. In a normal situation, this check gives a negative result, whereupon the sample counter of the product file is increased by one and the process continues by checking the number of samples again. The sample counter value is only increased after the product has passed through the weigher and the light screen. If the number of samples is still not full, the product identification data are stored in the learning file and the height register value is decreased by one, whereupon execution is transferred to the EAN block. On the other hand, if this check yields a full number of samples, the program calculates the average values in the learning file for the item in question, writes the product identification data into the product file, deletes the learning files, transmits the product identification data to the other checkout counters, decreases the height register value by one and proceeds to the EAN block. The transmission of the product identification data to the other checkout counters is shown in Fig. 2 as a "file updates" function, which transfers update data between the cash terminal program 15 and the product identification program 13. The data to be sent to the other checkout counters are transmitted via the cash terminal program 15 first to the background system 17 and via the background system's data transmission link 18 further to the other self-service checkout counters. In the event that the above-mentioned height check yields a negative result but the number of samples is full, an error situation is present because the system should already know all the correct height values. In this case, a possible cause of error is that the customer has not scanned the product but slid it along the conveyor belt 3 through the light screen, As a result, the system has not been able to read the EAN code of the product and the product identification program 13 reacts to this as if the number of samples were full. Also, there may be something wrong with the product height data. If this condition is true, the program jumps to the error block presented in Fig. 7, from where it returns to the scanning block as described before. While the system is learning the weight and shape data, the relevant data are stored in an auxiliary memory until the pr set numb r of samples is reached. After that, the average valu s of the data in question are saved in a product file in a permanent storage as describ d above.

After the product has passed through the light screen, it is considered as being accepted and execution is normally transferred from the learning block to the EAN block, presented in Fig. 6. First, the accepted EAN code is transmitted to the cash terminal program. The corr sponding function is shown in Fig. 2 as "accepted code". Next, the trough belt conveyor 7 is kept running for a certain length of time to ensure that the products are brought into the correct trough, the EAN register is decreased by one and a check is performed to ascertain whether the product arriving from the light screen was the last one. If it was not the last product, execution returns to the scanning block, whereas if it was the last product, the exit gate is opened for the customer, the belt conveyors are stopped after a suitable delay and the transaction is terminated. In addition, the receipt printer is given instructions to print out the receipt for the customer. This function is performed in Fig. 2 under "keyboard data", which is also the route used to transmit the termination signal to the cash terminal program. In addition to the manual termination signal, other information and messages may be typed in through a touch-screen, for example a request for help if the system does not function properly. In this case, the supervision personnel of the shop will assist the customer.

It is obvious to a person skilled in the art that the invention is not restricted to the examples described above, but that it may instead be varied within the scope of the following claims. Thus, far example the interface (14) agreed on by the suppliers of the product identification program (13) and the cash terminal program (15) may be a normal mechanical connection with pins or connector surfaces provided separately for each data type. In this case each program (13, 14) works separately, possibly even in different computers or processors.

Claims

1. Unstaffed checkout system, comprising a checkout counter (1) provided with actuators, a background system (17) containing at least the product data, and a cash terminal program (15) which transmits information from the checkout counter actuators to the background system, **characterized** in that
   - the system comprises a functional block (13) with an interface (14) compatible with the data transmitted by the cash terminal program (15),
   - data transmission between the functional block and the cash t rminal program takes place via the interface (14) in the functional block,
   - product data relating to product identification are collected in the functional

block and new data are automatically added to the data for each product until the system has learned the product identification data typical of each product.

2. Checkout system according to claim 1, **characterized** in that the functional block (13) installed in each checkout counter (1) is placed in a microcomputer (16) provided in each checkout counter so that the functional block (13) is linked as an extension to the existing cash system.

3. Checkout system according to claim 1 or 2, **characterized** in that the interface (14) between the functional block (13) installed in each checkout counter (1) and the cash terminal program (15) is an agreed variable, variable address or memory location in which the data collected and transmitted by the functional block is stored and from which the cash terminal program reads the data for subsequent further processing.

4. Procedure for accomplishing a transaction in an unstaffed checkout system as defined in claim 1, **characterized** in that the procedure comprises at least the following stages:
   - reading the credit card, accepting/rejecting the credit card and mode of payment for the checkout system
   - upon acceptance, starting the checkout counter, turning the distributor (8) and starting the scanning
   - checking the scanning result, weighing the products and recognition of product shape
   - after the weighing and shape recognition, comparing to the data obtained by scanning and to the data in the product file
   - as a result of the comparison, learning the product identification data and saving them in a storage if the system does not already have sufficient information about the product
   - accepting the sale of the product
   - conveying the product into a trough (11) for the customer to take away, opening the exit gate for the customer and stopping the operation of the checkout counter.

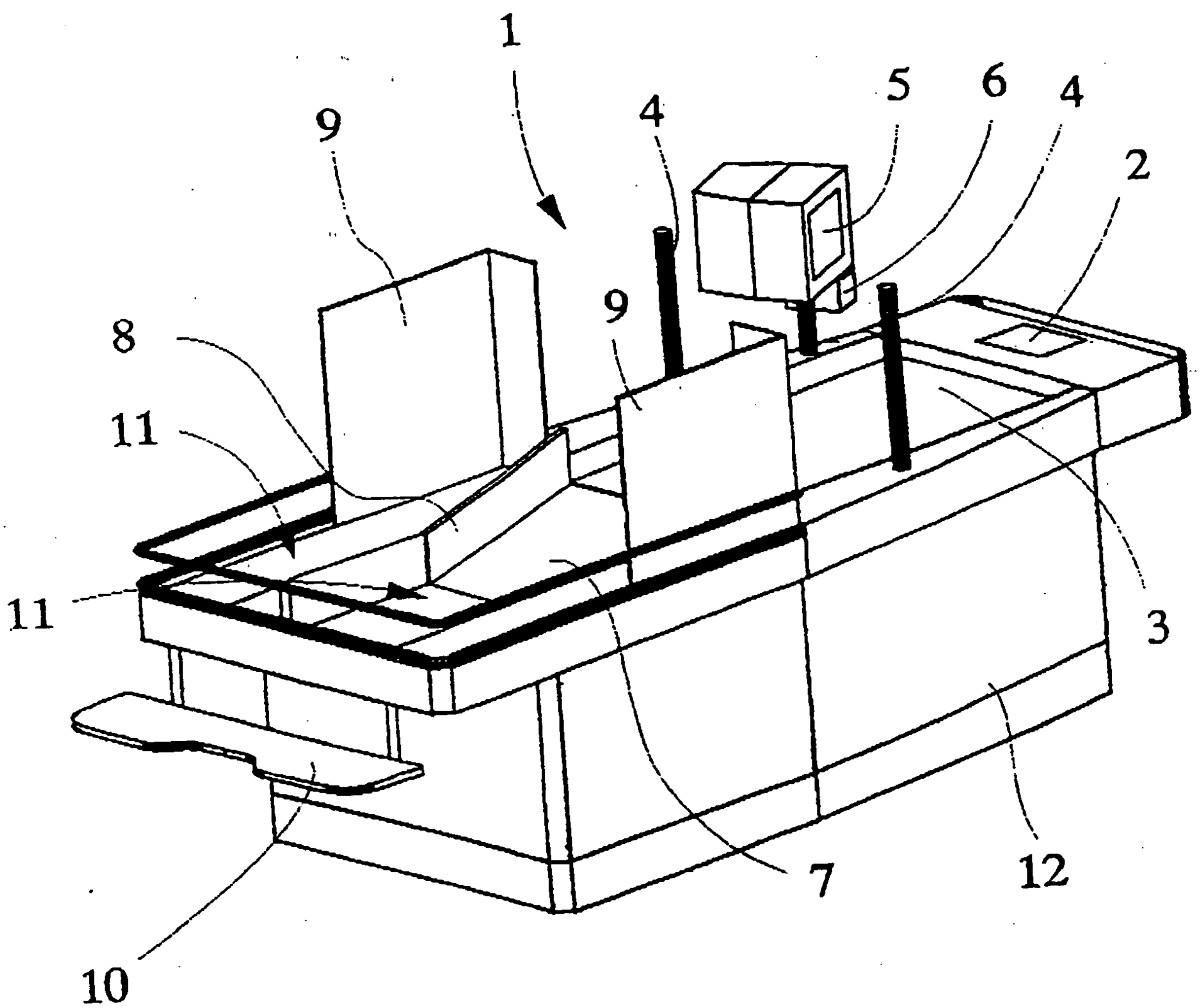

Fig. 1

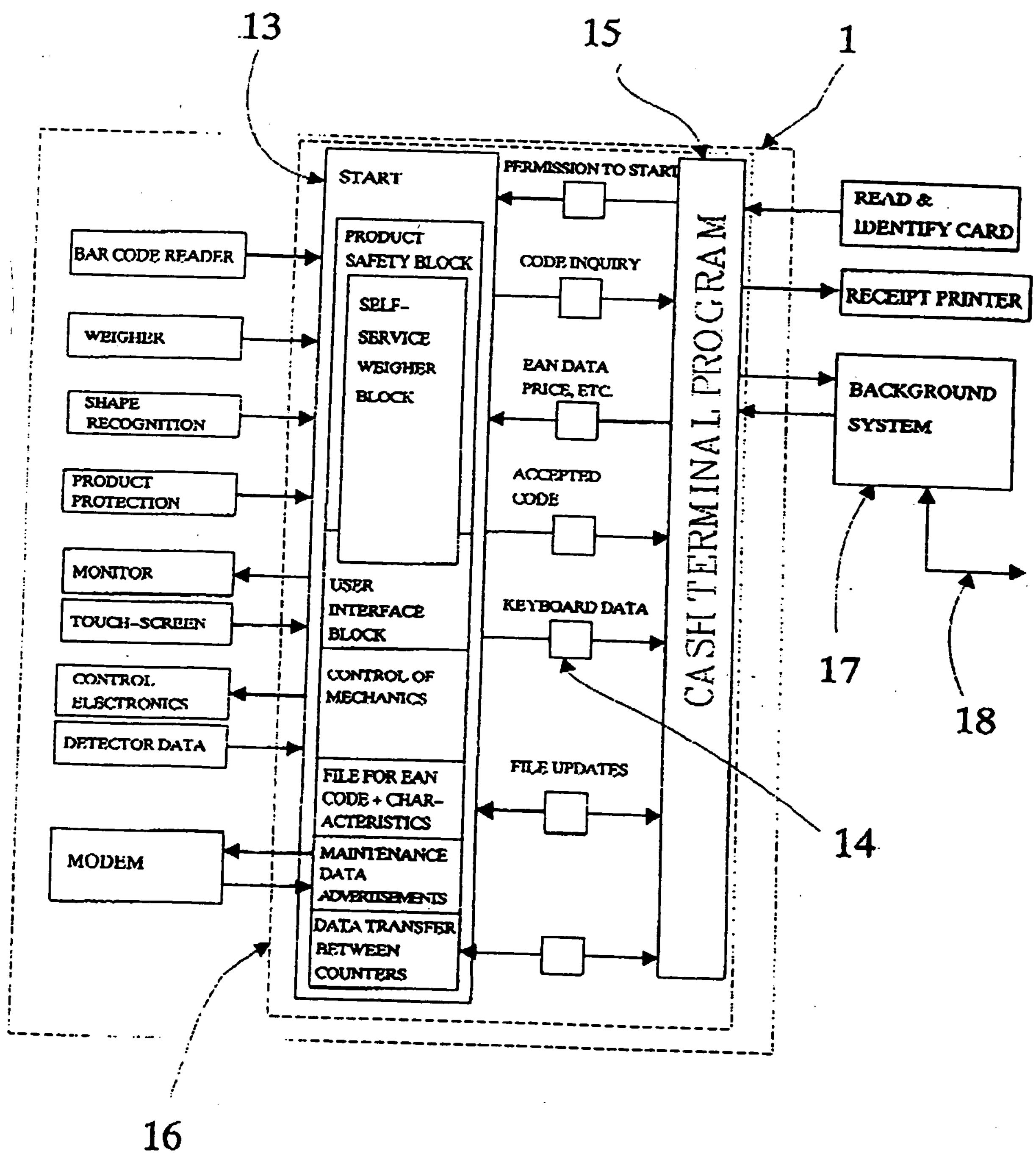
13
15
1
START
PERMISSION TO START
READ & IDENTIFY CARD
BAR CODE READER
PRODUCT SAFETY BLOCK
CODE INQUIRY
RECEIPT PRINTER
SELF-SERVICE WEIGHER BLOCK
WEIGHER
EAN DATA PRICE, ETC.
BACKGROUND SYSTEM
SHAPE RECOGNITION
PRODUCT PROTECTION
ACCEPTED CODE
CASH TERMINAL PROGRAM
MONITOR
USER INTERFACE BLOCK
KEYBOARD DATA
TOUCH-SCREEN
17
CONTROL ELECTRONICS
CONTROL OF MECHANICS
18
DETECTOR DATA
FILE UPDATES
FILE FOR EAN CODE + CHAR-ACTERISTICS
14
MODEM
MAINTENANCE DATA ADVERTISEMENTS
DATA TRANSFER BETWEEN COUNTERS
16

Fig. 2

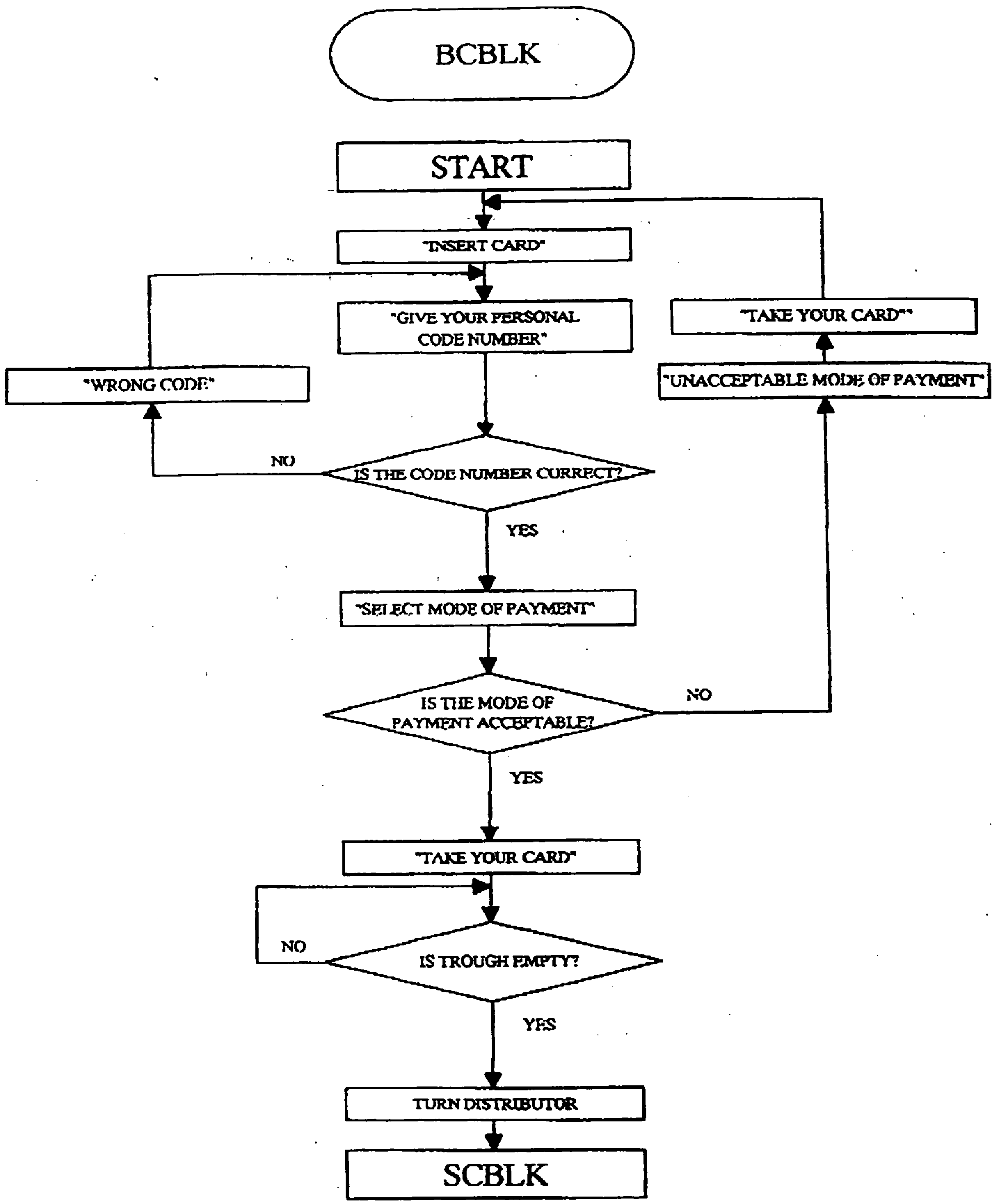
BCBLK
START
"INSERT CARD"
"GIVE YOUR PERSONAL CODE NUMBER"
"WRONG CODE"
NO
IS THE CODE NUMBER CORRECT?
YES
"SELECT MODE OF PAYMENT"
IS THE MODE OF PAYMENT ACCEPTABLE?
NO
"UNACCEPTABLE MODE OF PAYMENT"
"TAKE YOUR CARD"
YES
"TAKE YOUR CARD"
NO
IS TROUGH EMPTY?
YES
TURN DISTRIBUTOR
SCBLK

Fig. 3

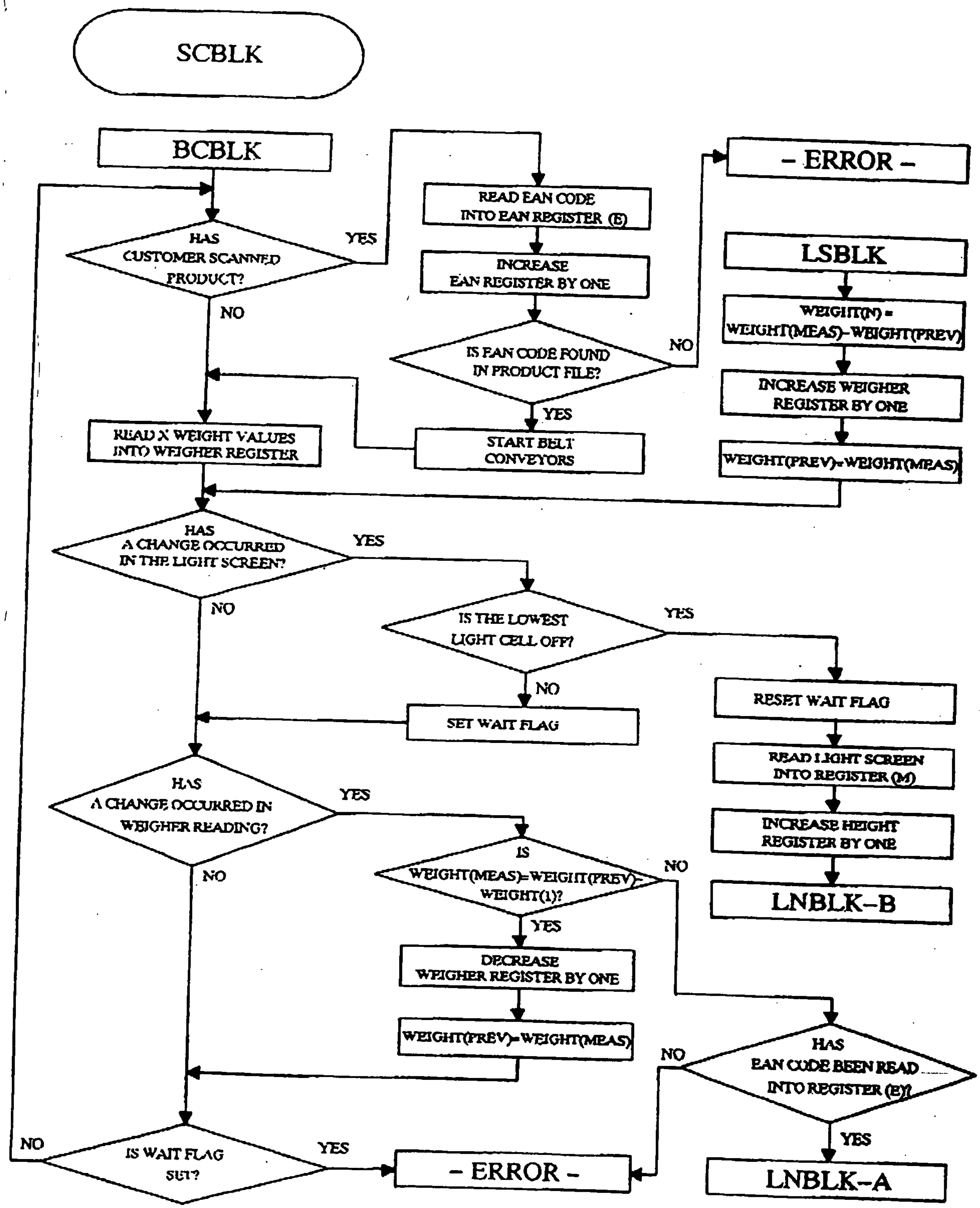
SCBLK
BCBLK
HAS CUSTOMER SCANNED PRODUCT?
YES
NO
READ EAN CODE INTO EAN REGISTER (E)
INCREASE EAN REGISTER BY ONE
IS EAN CODE FOUND IN PRODUCT FILE?
NO
YES
START BELT CONVEYORS
- ERROR -
LSBLK
WEIGHT(N) = WEIGHT(MEAS)-WEIGHT(PREV)
INCREASE WEIGHER REGISTER BY ONE
WEIGHT(PREV)=WEIGHT(MEAS)
READ X WEIGHT VALUES INTO WEIGHER REGISTER
HAS A CHANGE OCCURRED IN THE LIGHT SCREEN?
YES
NO
IS THE LOWEST LIGHT CELL OFF?
YES
NO
SET WAIT FLAG
RESET WAIT FLAG
READ LIGHT SCREEN INTO REGISTER (M)
INCREASE HEIGHT REGISTER BY ONE
LNBLK-B
HAS A CHANGE OCCURRED IN WEIGHER READING?
YES
NO
IS WEIGHT(MEAS)=WEIGHT(PREV)-WEIGHT(1)?
NO
YES
DECREASE WEIGHER REGISTER BY ONE
WEIGHT(PREV)=WEIGHT(MEAS)
HAS EAN CODE BEEN READ INTO REGISTER (E)?
NO
YES
LNBLK-A
NO
IS WAIT FLAG SET?
YES
- ERROR -

Fig. 4

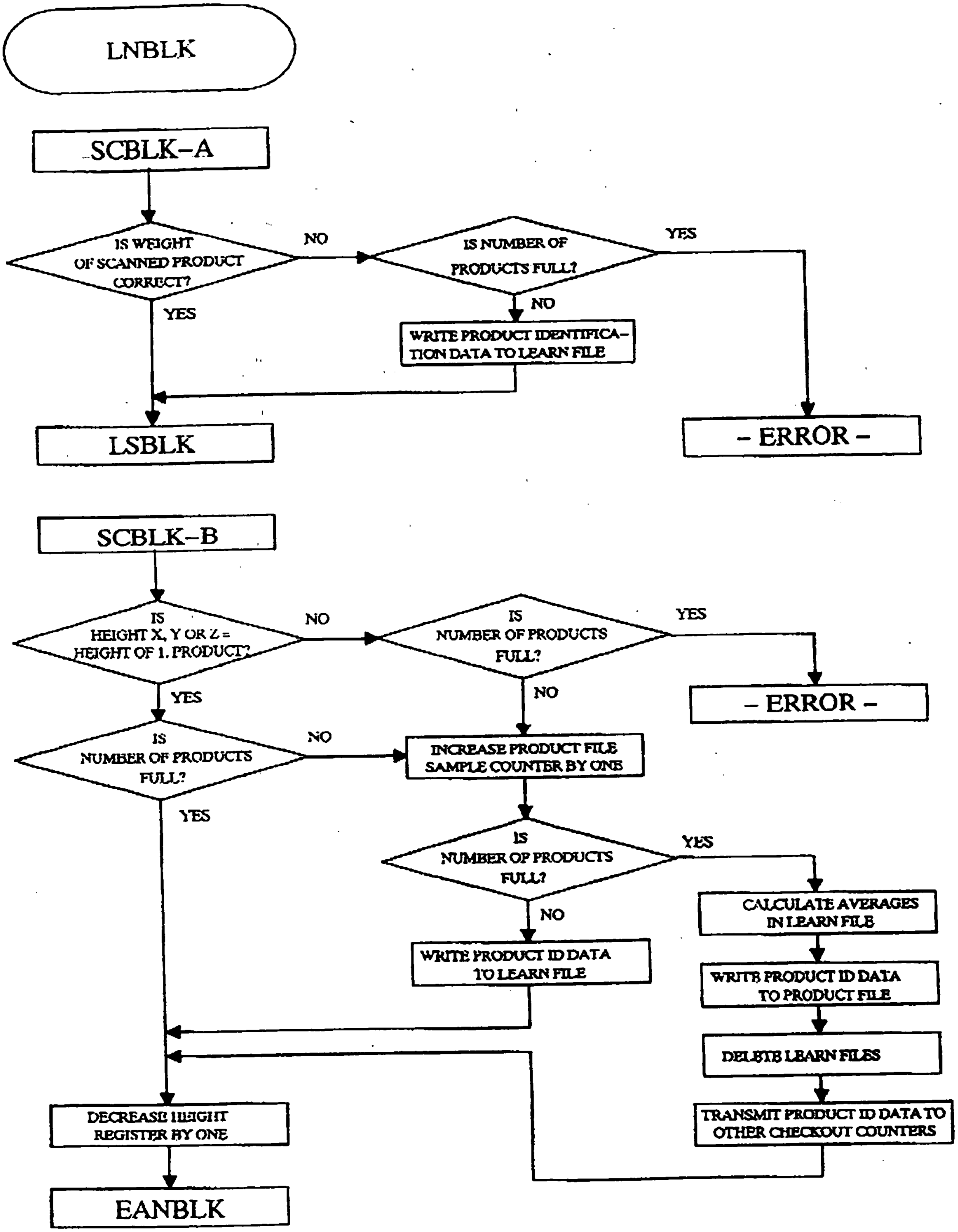
LNBLK
SCBLK-A
IS WEIGHT OF SCANNED PRODUCT CORRECT?
NO
IS NUMBER OF PRODUCTS FULL?
YES
NO
WRITE PRODUCT IDENTIFICATION DATA TO LEARN FILE
YES
LSBLK
- ERROR -
SCBLK-B
IS HEIGHT X, Y OR Z = HEIGHT OF 1. PRODUCT?
NO
IS NUMBER OF PRODUCTS FULL?
YES
NO
- ERROR -
YES
IS NUMBER OF PRODUCTS FULL?
NO
INCREASE PRODUCT FILE SAMPLE COUNTER BY ONE
YES
IS NUMBER OF PRODUCTS FULL?
YES
NO
CALCULATE AVERAGES IN LEARN FILE
WRITE PRODUCT ID DATA TO LEARN FILE
WRITE PRODUCT ID DATA TO PRODUCT FILE
DELETE LEARN FILES
TRANSMIT PRODUCT ID DATA TO OTHER CHECKOUT COUNTERS
DECREASE HEIGHT REGISTER BY ONE
EANBLK

Fig. 5

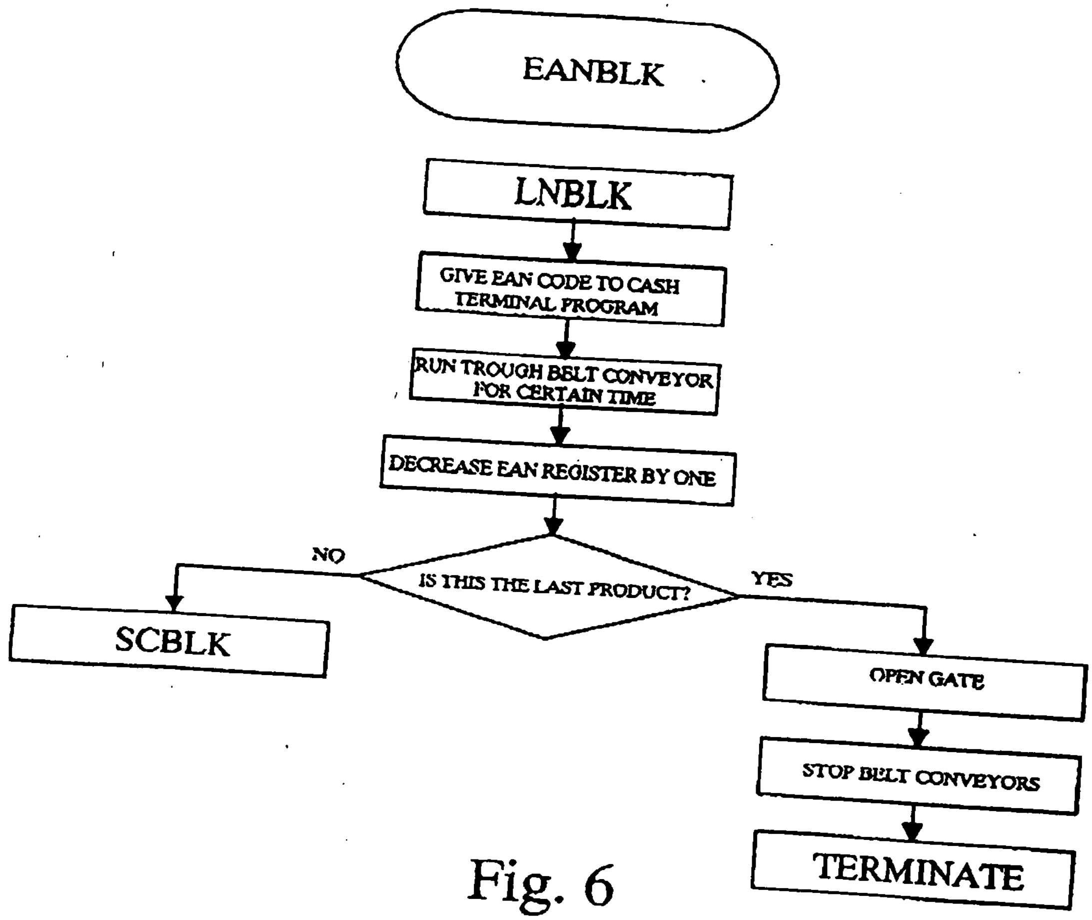
EANBLK
LNBLK
GIVE EAN CODE TO CASH TERMINAL PROGRAM
RUN TROUGH BELT CONVEYOR FOR CERTAIN TIME
DECREASE EAN REGISTER BY ONE
IS THIS THE LAST PRODUCT?
NO
YES
SCBLK
OPEN GATE
STOP BELT CONVEYORS
TERMINATE

Fig. 6

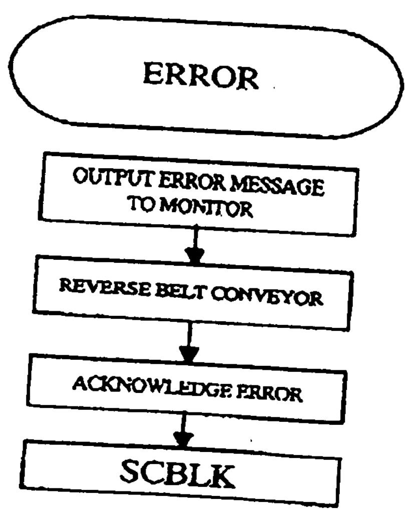
ERROR
OUTPUT ERROR MESSAGE TO MONITOR
REVERSE BELT CONVEYOR
ACKNOWLEDGE ERROR
SCBLK

Fig. 7